UniGeM: Unifying Data Mixing and Selection via Geometric Exploration and Mining

Anonymous ACL submission

Abstract

The scaling of Large Language Models (LLMs) is increasingly limited by data quality. Most methods handle data mixing and sample selection separately, which can break the structure in code corpora. We introduce **UniGeM**, a framework that unifies mixing and selection by treating data curation as a *manifold approximation* problem without training proxy models or relying on external reference datasets. UniGeM operates hierarchically: **Macro-Exploration** learns mixing weights with stability-based clustering; **Micro-Mining** filters high-quality instances by their geometric distribution to ensure logical consistency. Validated by training 8B and 16B MoE models on 100B tokens, UniGeM achieves **2.0× data efficiency** over a random baseline and further improves overall performance compared to SOTA methods in reasoning-heavy evaluations and multilingual generalization.

1 Introduction

The generalization of Large Language Models (LLMs) has traditionally relied on scaling parameters and data volume (Kaplan et al., 2020). However, recent shifts in scaling laws suggest that data quality now constrains model performance more than raw quantity (Hoffmann et al., 2022; Gadre et al., 2024). As high-quality public corpora are nearing depletion (Villalobos et al., 2022), simply aggregating noisy web data yields diminishing returns (Gunasekar et al., 2023). This motivates a focus on **data efficiency**: finding subsets that give more gain per compute (Li et al., 2023; Sorscher et al., 2022).

Most data efficiency approaches fall into two buckets: domain mixing and instance selection (Xie et al., 2023b; Yu et al., 2024; Liu et al., 2025). Data mixing serves as a **macro-distribution balancing** mechanism (Chen et al., 2023), but its coarse granularity often treats domains as flat distributions and overlooks structure within each domain (Diao et al., 2025). Meanwhile, many selection methods require training proxy models or reference datasets, which adds substantial overhead and may not track the target model's behavior at scale (Li et al., 2025; Yu et al., 2024; Li et al., 2024). Other selection methods, including heuristics and LLM-based scoring, evaluate samples independently and ignore the underlying manifold structure (Zhuang et al., 2025; Xie et al., 2023a). This decoupling leads to structural blind spots. We either optimize macro-distribution without considering micro-quality, or filter samples at the cost of disrupting the hierarchical dependencies. This matters most for the **code corpus**, a rigid logical manifold defined by brittle syntax and hierarchical dependencies (Li et al., 2023; Guo et al., 2020; Feng, 2020). We thus ask: *How can we unify macro-distribution balancing and micro-quality selection to identify the "golden subset" within a structured code corpus?*

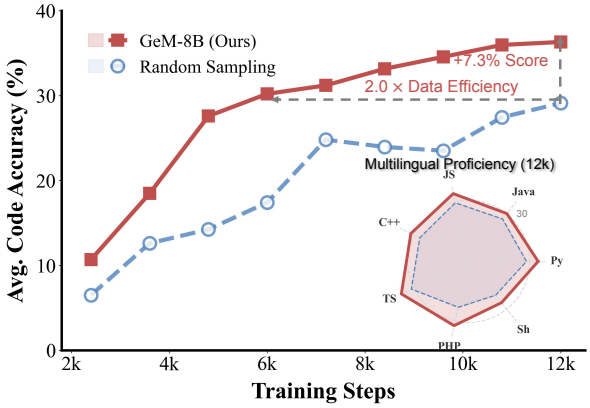

Figure 1: **Average downstream performance:** Random sampling vs. UniGeM for pre-training an 8B MoE model. The inset radar chart shows that UniGeM achieves stronger multilingual performance across programming languages.

To bridge this gap, we introduce **UniGeM (Geometric Exploration and Mining)**, a framework that unifies macro-distribution balancing and micro-quality selection. Unlike existing methods that rely on external reference datasets for alignment or re-

quire training expensive proxy models to estimate data importance, UniGeM views these tasks as a unified *manifold approximation* problem. First, **Macro-Exploration (Stage-I)** discovers semantic manifolds via stability-driven clustering to optimize macro-distribution balancing. Subsequently, **Micro-Mining (Stage-II)** performs micro-quality selection using intrinsic geometric priors to capture structural and logical dependencies. By measuring manifold deviations, UniGeM distills a **golden subset** that serves as a faithful approximation of the data manifold, preserving global topology and local dependency structure. We validate UniGeM by training 8B and 16B Mixture-of-Experts (MoE) models (both with 1.4B active parameters) from scratch on a 100B-token code-and-text mixture. Results show that UniGeM achieves superior data efficiency and model performance compared to existing baselines. Our main contributions are:

- **Unified Framework:** We propose a geometry-centric framework to unify macro-distribution balancing and micro-quality selection via manifold approximation.
- **Proven Data Efficiency:** UniGeM achieves $2.0\times$ **data efficiency** (vs. a random baseline) and performs better after one epoch.
- **Broader Gains:** UniGeM improves overall performance over SOTA baselines and shows stronger multilingual generalization.
- **Topological Insights:** Our ablations suggest that coverage and local structure matter for reasoning beyond per-sample quality scores, consistent with a manifold-approximation view of curation.

2 Proposed Method: The UniGeM Framework

In this section, we detail the implementation of UniGeM and provide a theoretical analysis to demonstrate effectiveness in data curation.

2.1 Problem Formulation and Overview

Let $\mathcal{D}_{raw} = \{x_i\}_{i=1}^{N}$ denote the uncurated corpus. We employ an embedding model f_θ to map each sample x_i to a normalized feature vector $\mathbf{e}_i \in \mathbb{S}^{d-1}$. We use this normalized space as a practical latent manifold $\mathcal{M}$ for clustering and selection. We cast data selection not as distributional alignment to an external reference, but as a self-contained **manifold approximation** problem. Our objective is to identify a subset $S \subset \mathcal{D}_{raw}$ that maximally preserves the topological structure of $\mathcal{M}$ while suppressing off-manifold outliers. Table 1 summarizes the key notations utilized throughout this framework.

Table 1: Summary of Key Notations and Symbols.

Symbol	Definition
I. Macro-Exploration (Stage-I)	
C_k	Global semantic cluster.
$\mathbf{z}_k, s_k$	Geometric features and score.
$\mathbf{w}$	Spectral consensus weights.
K^*	Optimal manifold resolution.
T_{scale}	Alignment scaling factor.
r_k	Global mixing budget.
II. Micro-Mining (Stage-II)	
S_j	Granular sub-cluster within C_k.
P_{S_j}	Semantic scores via probes.
$\mathcal{L}_{\text{struct}}$	Structural penalty.
β_{S_j}	Geometric cohesion gate.
$W(S_j)$	Final sampling weight.
III. Manifold Theory & Approximation	
$\mathcal{M}, d$	Latent manifold and dimensionality.
$\mathcal{E}(S)$	Wasserstein-2 approximation error.
σ_k^2	Intra-cluster variance of C_k in latent space.
α_k	Cluster mass (probability) $\mu(C_k)$.
$\Delta_{gain}^{(k)}$	Variance mass rejected by Stage-II pruning.

2.2 Stage-I: Macro Exploration (Global Clustering and Weighting)

As illustrated in **Figure 2 (Left)**, Stage-I performs the **Macro-Distribution Balancing**. This phase transforms the raw manifold into structured semantic clusters and assigns a sampling budget to each cluster. The process consists of three sequential steps:

Geometric Metrics and Scoring. First, we characterize each candidate cluster C_k via a feature vector $\mathbf{z}_k \in \mathbb{R}^4$. These dimensions act as geometric proxies:

- **Cohesion (z_{coh}):** Inverse intra-cluster distance. It prioritizes tight semantic structures akin to Neural Collapse (Papyan et al., 2020) to ensure gradient stability.
- **Cluster Size (z_{size}):** Sample volume. Used to mitigate head redundancy (e.g., boilerplate) and shift focus to the information-dense long tail (Huang et al., 2024).
- **Sequence Length (z_{len}):** The average token count, a proxy for verbosity, used to downweight overly long, low-information clusters.
- **Entropy (z_{ent}):** Distributional impurity of language identifiers. It penalizes semantic ambiguity to ensure domain purity.

To unify these proxies, we combine the normalized signals with a weighted linear score to derive

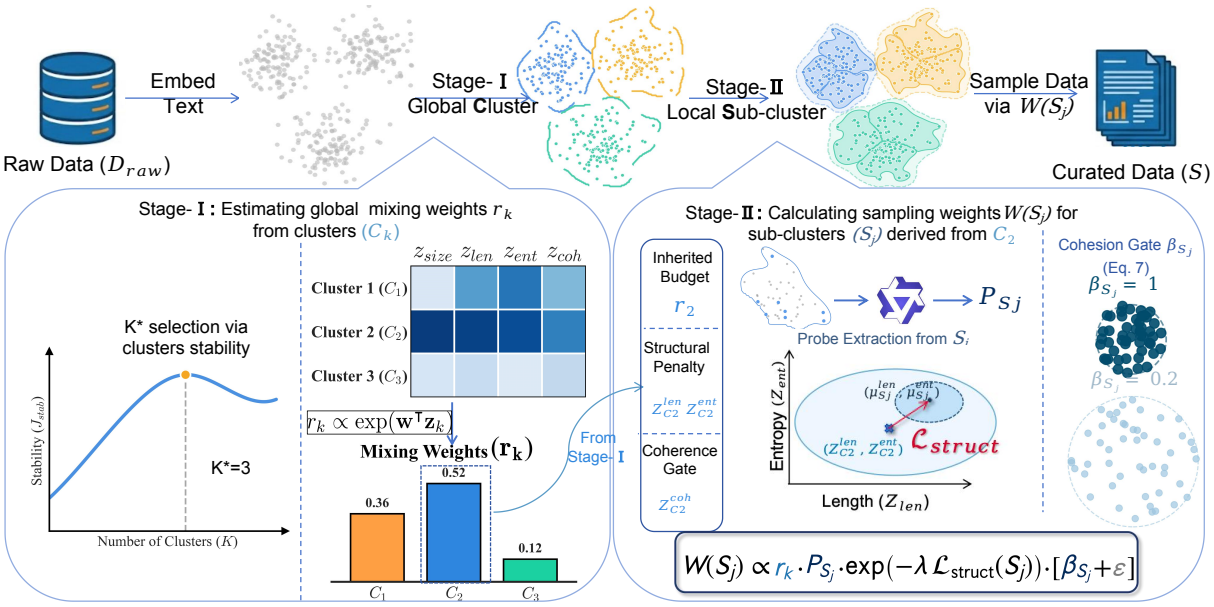

Figure 2: **Overview of the UniGeM framework.** The curation pipeline operates in two hierarchical stages: (Left) **Stage-I (Macro-Exploration)** identifies the intrinsic manifold resolution K^* via topological stability (Algorithm 1) and derives global mixing weights $\mathbf{r}$ through a softmax over geometric scores (Eq. 5), ensuring comprehensive global coverage across diverse semantic regions. (Right) **Stage-II (Micro-Mining)** computes sub-cluster sampling weights $W(S_j)$ by combining the inherited budget r_k, a semantic score modulated by a structural penalty $\exp(-\lambda \mathcal{L}_{struct}(S_j))$ (Eq. 6), and a cohesion gate $[\beta_{S_j} + \epsilon]$ (Eq. 7). This stage refines selection within each region to downweight off-manifold outliers while preserving representative local structural dependencies.

a scalar **Geometric Score** s_k. This score acts as a global quality metric, formulated as a contrast between structural coherence and statistical instability:

$$s_k = w_{\text{coh}} \tilde{z}_{\text{coh}} - \sum_{f \in \mathcal{F}_{neg}} w_f \tilde{z}_f \tag{1}$$

where $\mathcal{F}_{neg} = \{\text{len}, \text{ent}, \text{size}\}$ denotes the set of negative factors, and $\tilde{z}$ represents the Z-normalized magnitudes. The weighting vector $\mathbf{w}$ is derived via Spectral Consensus (see Appendix A).

Intrinsic Resolution Selection. Central to this is a **Cross-Resolution Soft Alignment**. We construct a probabilistic bridge $\pi_{K \to K'}$ based on centroid similarity:

$$\pi_{K \to K'}(j|i) = \text{Softmax}\left(T_{scale} \cdot \cos(\mu_i^{(K)}, \mu_j^{(K')})\right) \tag{2}$$

Using this bridge, we project the geometric scores $\mathbf{s}^{(K')}$ (derived in Eq. 1) from the finer resolution back to the current layer to obtain the **reconstructed scores** $\hat{\mathbf{s}}^{(K)}$:

$$\hat{s}_i^{(K)} = \sum_j \pi(j|i) s_j^{(K')} \tag{3}$$

Algorithm 1 Intrinsic Resolution via Soft-Alignment

Require: Manifold $\mathcal{M}$, Range $\mathcal{K}$, Stride ΔK.
Ensure: Optimal resolution K^*.

```
S ← ∅
for K ∈ K do
    K' ← K + ΔK
    1. Extract: Compute centroids C^(K) and scores s^(K).
    2. Align: Construct soft bridge π_{K→K'} (Eq. 2).
    3. Evaluate: Calculate rank stability J_stab(K) (Eq. 4).
    S ← S ∪ {(K, J_stab(K))}
end for
return K* = arg max_K {J : (K, J) ∈ S}
```

Stability is quantified via rank correlation using a Kendall's τ (detailed formulation in Appendix B):

$$J_{stab}(K) = \tau\left(\mathbf{s}^{(K)}, \hat{\mathbf{s}}^{(K)}\right) \tag{4}$$

The resolution K^* that maximizes this stability is selected as the intrinsic resolution (macro-granularity) via Algorithm 1.

Sampling Budget Allocation. With the optimal resolution K^* established, we finally allocate the

sampling budget $\mathbf{r}$. Leveraging the geometric scores from Eq. (1), the probability for each cluster is computed via a standard softmax:

$$r_k = \frac{\exp(s_k)}{\sum_{j=1}^{K^*} \exp(s_j)} \tag{5}$$

This framework ensures the budget concentrates on semantically dense regions (high s_k).

2.3 Stage-II: Micro-Mining (Local Sub-Clustering and Refinement)

As illustrated in Figure 2 (Right), Stage-II performs the **Micro-Quality Selection** via local sub-clustering. This phase decomposes each global cluster C_k into fine-grained sub-clusters $\{S_j\}$ to capture semantic diversity and topological details. The selection is refined through three coupled mechanisms:

Probe-based Semantic Scoring. To assess semantic content efficiently, we extract a small representative **Probe Set** (indicated by the solid centroids in Figure 2) from each sub-cluster S_j. We utilize an LLM as a **Knowledge Probe** to derive a **Semantic Score** P_{S_j}. Implementation details are provided in Appendix D.

Relative Structural Consistency. To enforce manifold consistency, we define a **Structural Penalty** $\mathcal{L}_{struct}$ using a rectified Mahalanobis distance. It penalizes sub-clusters S_j where the empirical mean $z_f(S_j)$ of features (e.g., length, entropy) exceeds the geometric consensus (μ_f, σ_f) of the parent cluster C_k:

$$\mathcal{L}_{struct} = \sum_{f \in \{len, ent\}} \left[\frac{z_f^{(S_j)} - \mu_f^{(C_k)}}{\sigma_f^{(C_k)}} \right]_+^2 \tag{6}$$

We introduce $\mathcal{L}_{\text{struct}}$ not only to ensure that the curated subset preserves the underlying manifold topology but also to mitigate the score saturation observed in semantic scores (Fig. 10 and Fig. 11).

Geometric Cohesion Gate. To ensure topological robustness, we apply a **Geometric Cohesion Gate** β_{S_j}, depicted as the gating module in Figure 2. Leveraging the cohesion metric z_{coh} defined in Stage-I, we modulate sampling confidence based on the sub-cluster's compactness relative to its parent cluster:

$$\beta_{S_j} = \text{Sigmoid}\left(z_{coh}^{(S_j)} - z_{coh}^{(C_k)} \right) \tag{7}$$

This gating mechanism suppresses sub-clusters with lower cohesion than their parent ($z_{coh}^{(S_j)} < z_{coh}^{(C_k)}$) while retaining structurally compact regions.

Hierarchical Sampling Weight. The final sampling probability $W(S_j)$ synthesizes the global budget r_{C_k} with local metrics via a multiplicative modulation. It is computed as:

$$W(S_j) \propto \underbrace{r_k}_{\text{Budget}} \cdot \underbrace{P_{S_j} \exp\left(-\lambda \mathcal{L}_{struct}(S_j)\right)}_{\text{Geometry-Aware Score}} \cdot \underbrace{[\beta_{S_j} + \epsilon]}_{\text{Gate}} \tag{8}$$

Within the geometry-aware score, we use a semantic score modulated by a geometric penalty to suppress off-manifold sub-clusters. Simultaneously, the gate $\left[\beta_{S_j} + \epsilon\right]$ ensures cohesion, with ϵ preventing mode collapse by maintaining a minimal exploration floor.

2.4 Theoretical Analysis: Manifold Approximation

We frame data selection as minimizing the transport cost between the empirical measure $\hat{\mu}_S$ and the true manifold distribution μ (Villani, 2009). Let $f_\theta : \mathcal{X} \to \mathbb{R}^d$ be an L-Lipschitz embedding into a latent manifold of intrinsic dimension $d \ll D$ (Pope et al., 2021; Du et al., 2021).

Definition 1 (Approximation Error). We quantify the quality of subset S by the squared Type-2 Wasserstein distance (Peyré et al., 2019):

$$\begin{aligned} \mathcal{E}(S) &\triangleq W_2^2(\mu, \hat{\mu}_S) \\ &= \inf_{\gamma \in \Pi(\mu, \hat{\mu}_S)} \iint_{\mathcal{M}^2} \|x - y\|^2 d\gamma(x, y). \end{aligned} \tag{9}$$

A constructive two-stage transport argument (Appendix C) yields the following constant-factor decomposition:

$$\begin{aligned} \mathcal{E}(S) \leq 2 \underbrace{\sum_{k=1}^{K} \int_{C_k} \|x - \mathbf{c}_k\|^2 \, d\mu(x)}_{\text{Stage-I: Quantization}} \\ + 2 \underbrace{\sum_{k=1}^{K} \alpha_k \mathbb{E}_{x \sim \hat{\mu}_{S_k}} \|x - \mathbf{c}_k\|^2}_{\text{Stage-II: Pruning}}. \end{aligned} \tag{10}$$

Theorem 1 (UniGeM Bound). For density $p(x)$, the UniGeM approximation error satisfies:

$$\mathcal{E}(S_{UniGeM}) \leq 2C_d K^{-2/d} + 2 \sum_{k=1}^{K} \alpha_k \left(\sigma_k^2 - \Delta_{gain}^{(k)}\right), \tag{11}$$

where C_d is the Zador constant (Zador, 1982), $\alpha_k \triangleq \mu(C_k)$, and $\sigma_k^2 \triangleq \mathbb{E}_{x\sim\mu(\cdot|C_k)}\|x-\mathbf{c}_k\|^2$. Here $\Delta_{gain}^{(k)} \geq 0$ captures the within-cluster reduction induced by Stage-II pruning. $\mathcal{L}_{struct}$ is a practical signal for identifying the outliers driving $\Delta_{gain}^{(k)}$.

Remark. The bound is derived under idealized modeling assumptions and is intended to guide the design rather than exactly characterize all engineering details.

3 Experimental Setup

We evaluate UniGeM on **large-scale code pretraining**. Code is a demanding setting for data curation: small syntax issues can break execution (Li et al., 2023), the corpus spans many programming languages (Feng, 2020), and programs follow hierarchical dependencies (Guo et al., 2020). These properties make naive filtering brittle, and they let us test whether UniGeM preserves both coverage and local structure.

3.1 Corpus Construction and Sampling Protocol

We construct a 100B-token training corpus by mixing **The Stack Dedup** (Kocetkov et al., 2022) and Common Crawl (Schäfer, 2017) in a fixed 7:3 code-to-text ratio. The 30B text component is frozen across all experiments. This fixed ratio ensures that gains on code benchmarks stem from UniGeM's strategic data blending rather than a simple inflation of total code volume (Xie et al., 2023b).

3.2 UniGeM Implementation Details

Deploying geometric clustering on terabytes of data presents a computational challenge. We adopt a *Probing-and-Scaling* strategy:

- **Phase I: Macro-Topology Discovery.** We first removed near-duplicates of evaluation benchmarks from the raw corpus. Subsequently, we embed the corpus using *Qwen3-embedding* (Zhang et al., 2025b). To ensure scalability, we perform K-means clustering on a representative subset ($\mathcal{D}_{probe} \approx 20\%$) to identify the optimal macro-granularity $K^* = 72$. Each sample in the full corpus is assigned to its nearest global centroid.
- **Phase II: Hierarchical Geometric Mining.** Within each global cluster C_k, we execute localized sub-clustering to capture fine-grained micro-structures. The sub-cluster density S_j is determined by the square root of the cluster population.

3.3 Model and Hyperparameter

We adopt a fine-grained sparse Mixture-of-Experts (MoE) Transformer, increasing the total expert capacity while keeping the number of activated parameters per token fixed to preserve inference efficiency (Liu et al., 2024). We instantiate **UniGeM-8B** (32 experts) and **UniGeM-16B** (64 experts), both with **1.4B** active parameters. All models are trained **from scratch** with the same training recipe; only the data curation differs [1]. For geometric selection, we use $\lambda = 0.5$, $T_{scale} = 20$, and $\epsilon = 0.01$. See Appendix E for details.

3.4 Compared Methods

Baseline & SOTA Methods

- **Random Sampling:** Uniform sampling from the raw corpus as a reference baseline.
- **Meta-rater** (Zhuang et al., 2025): A representative *instance-level selection* method relying on *LLM-based scoring* to rank individual data quality.
- **Nemotron-CLIMB** (Diao et al., 2025): A representative *domain-level data mixing* method that optimizes global balancing weights.

To ensure competitive baselines, we apply a **code adaptation** to Meta-rater and CLIMB to reduce domain mismatch (Appendix E).

Ablation Variants To validate the hierarchical design, we compare UniGeM against:

- **Cluster Random:** Replaces the coarse-to-fine hierarchy with a single flat clustering stage ($K = K_{total}$) and performs random sampling with a uniform ratio in each cluster.
- ***w/o* Hierarchy:** Uses the same flat clustering structure as above ($K = K_{total}$) but selects data via UniGeM's micro-mining metrics instead of random sampling.
- ***w/o* Stage-I:** Assigns uniform global budgets ($r_k = 1/K$) while retaining Stage-II mining, utilizing **global feature statistics** as the reference for both the structural and the geometric cohesion gate.
- ***w/o* Stage-II:** Uses optimized global weights $\mathbf{r}$ but performs random sampling within clusters.

[1] Reproducibility statement is discussed in Appendix H.

Table 2: Coding benchmark results. We compare the scaling properties between UniGeM-16B and UniGeM-8B under a constant inference budget (1.4B active parameters). The best scores in Block 2 and Block 3 are underlined.

Setting		Avg.	HE	HE$^+$	MBPP	MBPP$^+$	LiveCode	CruxEval	
Methods	**Epoch**	**Score**	**Pass@1**	**Pass@1**	**Pass@1**	**Pass@1**	**Pass@1**	**Input**	**Output**
UniGeM-16B (16B Total / 1.4B Active)									
Random	1.0	32.9	55.9	48.6	33.5	37.6	1.1	29.8	24.1
UniGeM (Ours)	0.5	32.3 ↓0.6	49.9 ↓6.0	43.8 ↓4.8	34.6 ↑1.1	40.9 ↑3.3	6.4 ↑5.3	26.0 ↓3.8	24.6 ↑0.5
UniGeM (Ours)	1.0	39.5 ↑6.6	59.3 ↑3.4	55.1 ↑6.5	41.8 ↑8.3	47.8 ↑10.2	10.8 ↑9.7	31.6 ↑1.8	30.1 ↑6.0
UniGeM-8B (8B Total / 1.4B Active)									
Random	1.0	29.1	45.3	44.1	30.5	34.2	2.0	22.3	25.5
Meta-rater	1.0	35.0 ↑5.9	53.2 ↑7.9	49.6 ↑5.5	35.3 ↑4.8	43.5 ↑9.3	4.7 ↑2.7	31.4 ↑9.1	27.7 ↑2.2
CLIMB	1.0	35.2 ↑6.1	52.4 ↑7.1	48.8 ↑4.7	36.7 ↑6.2	45.5 ↑11.3	6.9 ↑4.9	29.7 ↑7.4	26.6 ↑1.1
UniGeM (Ours)	0.5	30.0 ↑0.9	47.7 ↑2.4	43.5 ↓0.6	32.4 ↑1.9	42.2 ↑8.0	5.2 ↑3.2	16.6 ↓5.7	22.5 ↓3.0
UniGeM (Ours)	1.0	36.4 ↑7.3	53.7 ↑8.4	50.3 ↑6.2	37.4 ↑6.9	46.6 ↑12.4	7.8 ↑5.8	31.0 ↑8.7	28.4 ↑2.9
Ablation Study (based on UniGeM-8B)									
Random	1.0	29.1	45.3	44.1	30.5	34.2	2.0	22.3	25.5
Cluster Random	1.0	29.2 ↑0.1	46.0 ↑0.7	44.1	31.1 ↑0.6	34.6 ↑0.4	1.8 ↓0.2	21.4 ↓0.9	25.2 ↓0.3
w/o Hierarchy	1.0	33.0 ↑3.9	51.4 ↑6.1	48.1 ↑4.0	34.5 ↑4.0	40.1 ↑5.9	1.9 ↓0.1	27.0 ↑4.7	28.0 ↑2.5
w/o Stage-II	1.0	31.2 ↑2.1	45.8 ↑0.5	43.8 ↓0.3	33.8 ↑3.3	40.1 ↑5.9	2.3 ↑0.3	24.3 ↑2.0	28.3 ↑2.8
w/o Stage-I	1.0	34.9 ↑5.8	52.2 ↑6.9	49.5 ↑5.4	36.1 ↑5.6	43.1 ↑8.9	4.5 ↑2.5	31.3 ↑9.0	27.5 ↑2.0
UniGeM (Ours)	1.0	36.4 ↑7.3	53.7 ↑8.4	50.3 ↑6.2	37.4 ↑6.9	46.6 ↑12.4	7.8 ↑5.8	31.0 ↑8.7	28.4 ↑2.9

3.5 Code Benchmarks

We evaluate UniGeM on a diverse set of coding benchmarks covering code generation, robustness/reasoning, and multilingual proficiency:

- **Code Generation:** We utilize *HumanEval(+)* (**HE**) (Chen, 2021) and *MBPP(+)* (Austin et al., 2021) to evaluate generation performance.
- **Robustness & Reasoning:** We assess temporal generalization using *LiveCodeBench* (**LCB**) (Jain et al., 2024) and execution prediction capabilities using *CruxEval* (Gu et al., 2024).
- **Multilingual Proficiency:** We conduct a fine-grained analysis across multiple programming languages using *MultiPL-E* (Cassano et al., 2023).

4 Result Analysis

In this section, we evaluate UniGeM from multiple perspectives. We compare it with state-of-the-art baselines, examine cross-lingual generalization, and study the impact of the hierarchical design and training efficiency through ablation and scaling analyses.

4.1 Comparison with State-of-the-Art

We compare UniGeM against the Random baseline, **CLIMB** and **Meta-rater** on the 8B MoE model. Table 2 summarizes the results; the **Avg.** score represents the unweighted arithmetic mean of all reported metrics.

Overall Superiority and Data Efficiency. UniGeM achieves the best overall score (**36.4**), above the adapted domain-mixing baseline (**CLIMB**, 35.2) and the adapted instance-selection baseline (**Meta-rater**, 35.0). It is also more data-efficient: UniGeM reaches **30.0** after 0.5 epochs, surpassing Random sampling at 1.0 epoch (29.1), corresponding to an approximate 2.0× efficiency gain. After 1.0 epoch, UniGeM further improves to **36.4**, indicating that the gain is not limited to early training.

Reasoning and Generalization. UniGeM's advantage is particularly robust on complex execution and out-of-distribution (OOD) tasks. On

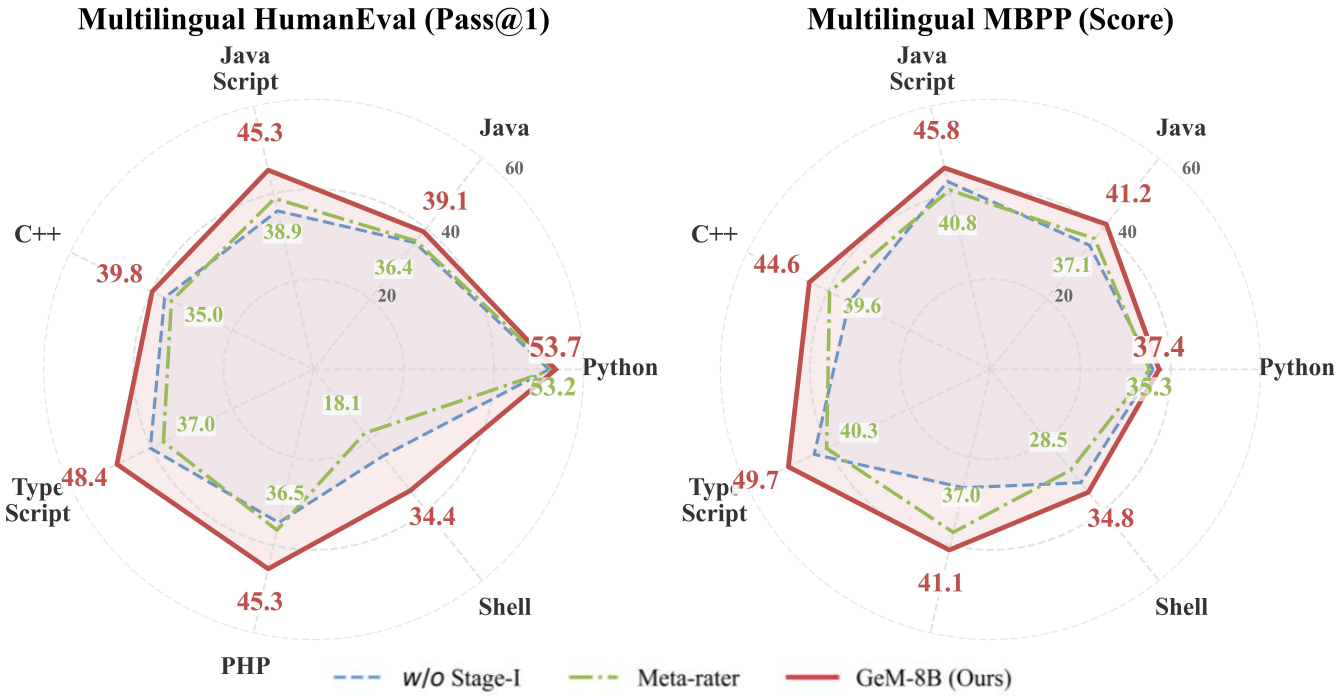

Figure 3: **Impact of Macro-Exploration (Stage-I).** UniGeM (red) outperforms Meta-rater (green) and the *w/o Stage-I* variant (blue) across 7 languages, indicating that global clustering improves multilingual generalization.

LiveCodeBench, UniGeM scores **7.8**, clearly surpassing CLIMB (6.9) and significantly exceeding Meta-rater (4.7). On **CruxEval**, the adapted Meta-rater remains highly competitive and slightly leads on **Input** prediction, suggesting that code-aware LLM scoring captures common logical patterns. UniGeM performs best on **Output** execution prediction (**CruxEval-Output**) with **28.4**, indicating better coverage of more complex execution behaviors.

4.2 Multilingual Proficiency

We evaluate cross-lingual generalization on **Multilingual HumanEval** and **MBPP** (Fig. 3). UniGeM consistently dominates the Meta-rater and *w/o Stage-I* variant across 7 languages. On HumanEval, UniGeM achieves decisive leads in low-resource domains like **Shell** (**34.4%** vs. Meta's 18.1%) and **TypeScript** (**48.4%** vs. 37.0%), while maintaining clear advantages in strict-syntax languages like **C++** (**39.8%** vs. 35.0%) and **Java** (**39.1%** vs. 36.4%). This superiority persists on MBPP (e.g., **44.6%** vs. 39.6% in C++), confirming UniGeM's robust, language-agnostic transfer capabilities.

4.3 Ablation and Hyperparameter Analysis

We conduct ablations and sensitivity analyses on UniGeM-8B to isolate the contribution of each component.

Contribution of Model Components. Ablations highlight the role of each stage. Removing Stage-I (*w/o Stage-I*) replaces learned global budgets with uniform allocation, which weakens global coverage; performance stays relatively stable on high-resource languages (e.g., Python) but drops on under-represented languages such as Shell and PHP in multilingual evaluation. Conversely, removing Stage-II (*w/o Stage-II*) reduces HumanEval Pass@1 from 53.7% to 45.8%, indicating that local geometric refinement is important even with optimized macro mixing.

Statistical Properties of Stage-I Features. Statistical analysis confirms that extensive features (e.g., length) follow a Log-Normal distribution while intensive features (e.g., cohesion) are naturally stable, supporting our hybrid geometric priors (details in Appendix F).

Sensitivity to Global Granularity (K). The cluster number K controls the resolution of the global approximation. As shown in Figure 4, the stability index $J_{stab}(K)$ increases and then plateaus beyond $K \approx 60$. We choose $K^* = 72$ as a stable setting with finer semantic resolution; smaller K (e.g., $K < 40$) suffers from under-segmentation, merging distinct semantic domains and reducing stability. Beyond K^*, the marginal gain (ΔJ) diminishes to near zero, suggesting that

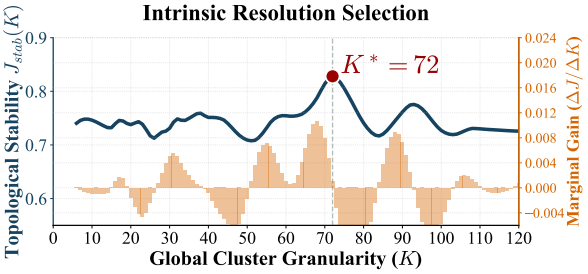

Figure 4: **Intrinsic Resolution Selection.** The stability index $J_{\text{stab}}(K)$ (blue line) rises and then plateaus, attaining its maximum at $K^* = 72$. The marginal gain (orange bars) diminishes significantly beyond this point.

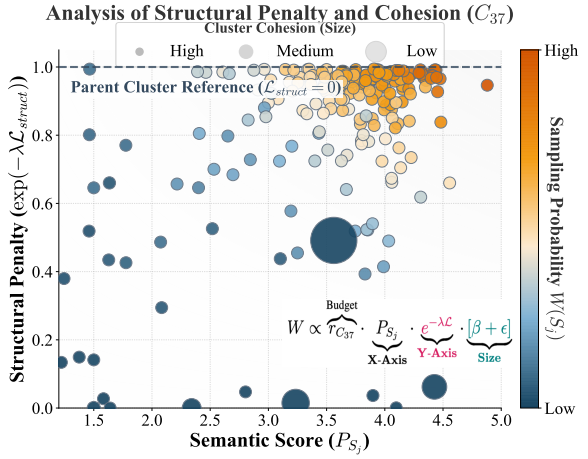

Figure 5: Empirical distribution of a representative subset of sub-clusters in C_{37}. Sub-clusters are mapped by **Semantic Score** (P_{S_j}, X-axis) and **Structural Penalty** (Y-axis), with bubble size representing **Cluster Cohesion**. The color gradient indicates the **Sampling Probability** $W(S_j)$, illustrating the budget allocation toward sub-clusters.

72 clusters are sufficient to capture the corpus's semantic structure without over-segmentation or fragmentation.

Analysis of Structural Penalty and Cohesion. Figure 5 illustrates how UniGeM balances semantic utility and structural consistency at the sub-cluster level. Sub-clusters with higher **Semantic Score** and a small structural deviation (i.e., **Structural Penalty** close to 1) receive higher sampling probability, whereas sub-clusters that score well semantically but deviate structurally are downweighted. **Cluster Cohesion** (bubble size) further biases sampling toward compact, well-formed sub-clusters that better represent the parent cluster.

4.4 Scaling Analysis

Model Scaling Dynamics. We further study scaling by training a larger **UniGeM-16B** model. At 0.5 epochs, UniGeM-16B reaches **32.3**, which is lower than its 1.0-epoch score, consistent with larger models being more data-hungry (Hoffmann et al., 2022). After 1.0 epoch, UniGeM-16B improves to **39.5** (**+6.6** over Random), showing that the curated corpus continues to benefit training as model size increases.

5 Related Work

Current work on data efficiency is moving beyond static filtering toward methods that model dataset structure. One prominent line trains **proxy models** to derive mixing or importance weights (e.g., DoReMi (Xie et al., 2023b) and DCLM (Li et al., 2024)), but this can be compute-heavy and may introduce "proxy bias," where signals from small proxies do not transfer cleanly to larger target models (Mindermann et al., 2022; Sorscher et al., 2022). A related family relies on reference datasets for alignment or selection (Xie et al., 2023a; Li et al., 2025), while model-aware approaches such as Mates (Yu et al., 2024; Zhang et al., 2025a) use influence-style estimates to capture sample-level contributions. To better preserve reasoning ability, QuaDMix (Liu et al., 2025) explicitly balances quality and diversity, though gradient- or training-intensive signals can limit scalability (Mindermann et al., 2022; Li et al., 2024). More principled directions aim to avoid expensive training signals altogether: DDOQ (Tan and Slade, 2025) casts selection as pushforward optimal quantization, improving over heuristic clustering schemes (Chen et al., 2023; Diao et al., 2025), and Wasserstein-manifold views model dataset dynamics in a way that goes beyond flat domain mixing toward preserving structure relevant for complex reasoning (Atanackovic et al., 2024).

6 Conclusion

We introduced **UniGeM**, a hierarchical framework that unifies macro-distribution balancing and micro-quality selection through *manifold approximation*. By using topological stability to choose the global resolution and geometric priors for instance mining, UniGeM curates a compact, structure-preserving training set from code corpora. Experiments with 8B and 16B MoE models show **2.0× data efficiency** over a random baseline and **better one-epoch performance** than strong adapted baselines, with consistent gains in code reasoning and multilingual evaluations.

Limitations

Despite its effectiveness, this work has several limitations:

1. **Domain Specificity:** Our evaluation focused primarily on the **code corpus**. While code provides a rigorous testbed for geometric structures, the efficacy of UniGeM on massive, heterogeneous **general web text** mixtures remains to be fully explored.
2. **Computational Overhead:** The initial global embedding and clustering phase, while mitigated by our probing-and-scaling strategy, still requires non-trivial resources when applied to trillion-token scales.
3. **Static Pipeline:** The current framework operates as a pre-processing step. Future work is required to integrate UniGeM into online training pipelines to allow for dynamic, cross-domain manifold updates as the model's data needs evolve.

References

Lazar Atanackovic, Xi Zhang, Brandon Amos, Mathieu Blanchette, Leo J Lee, Yoshua Bengio, Alexander Tong, and Kirill Neklyudov. 2024. Meta flow matching: Integrating vector fields on the wasserstein manifold. *arXiv preprint arXiv:2408.14608*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, and 1 others. 2023. Multipl-e: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 49(7):3675–3691.

Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2023. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36:36000–36040.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, and 1 others. 2025. Climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. *arXiv preprint arXiv:2504.13161*.

Yilun Du, Katie Collins, Josh Tenenbaum, and Vincent Sitzmann. 2021. Learning signal-agnostic manifolds of neural fields. *Advances in Neural Information Processing Systems*, 34:8320–8331.

Z Feng. 2020. Codebert: A pre-trained model for program-ming and natural languages. *arXiv preprint arXiv:2002.08155*.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, and 1 others. 2024. Language models scale reliably with over-training and on downstream tasks. *arXiv preprint arXiv:2403.08540*.

Robert M. Gray and David L. Neuhoff. 2002. Quantization. *IEEE transactions on information theory*, 44(6):2325–2383.

Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. 2024. Cruxeval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, and 1 others. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie Liu, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, and 1 others. 2020. Graphcodebert: Pre-training code representations with data flow. *arXiv preprint arXiv:2009.08366*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *Advances in Neural Information Processing Systems*, 35:30016–30030.

Jing Huang, Diyi Yang, and Christopher Potts. 2024. Demystifying verbatim memorization in large language models. *arXiv preprint arXiv:2407.17817*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, and 1 others. 2022. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

Michel Ledoux. 2001. *The concentration of measure phenomenon*. 89. American Mathematical Soc.

Jiazheng Li, Lu Yu, Qing Cui, Zhiqiang Zhang, Jun Zhou, Yanfang Ye, and Chuxu Zhang. 2025. MASS: Mathematical data selection via skill graphs for pretraining large language models. *arXiv preprint arXiv:2503.14917*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, and 1 others. 2023. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*.

Ruihang Li, Yixuan Wei, Miaosen Zhang, Nenghai Yu, Han Hu, and Houwen Peng. 2024. Scalingfilter: Assessing data quality through inverse utilization of scaling laws. *arXiv preprint arXiv:2408.08310*.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Fengze Liu, Weidong Zhou, Binbin Liu, Zhimiao Yu, Yifan Zhang, Haobin Lin, Yifeng Yu, Bingni Zhang, Xiaohuan Zhou, Taifeng Wang, and 1 others. 2025. Quadmix: Quality-diversity balanced data selection for efficient llm pretraining. *arXiv preprint arXiv:2504.16511*.

Sören Mindermann, Jan M Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and 1 others. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt (rho-loss). In *International Conference on Machine Learning*, pages 15630–15660. PMLR.

Vardan Papyan, XY Han, and David L Donoho. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663.

Gabriel Peyré, Marco Cuturi, and 1 others. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. 2021. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations (ICLR)*.

Roland Schäfer. 2017. Accurate and efficient general-purpose boilerplate detection for crawled web corpora. *Lang. Resour. Eval.*, 51(3):873–889.

ByteDance Seed, Yuyu Zhang, Jing Su, Yifan Sun, Chenguang Xi, Xia Xiao, Shen Zheng, Anxiang Zhang, Kaibo Liu, Daoguang Zan, and 1 others. 2025. Seed-coder: Let the code model curate data for itself. *arXiv preprint arXiv:2506.03524*.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Morcos Ariyo. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536.

Hong Ye Tan and Emma Slade. 2025. Dataset distillation as pushforward optimal quantization. *arXiv preprint arXiv:2501.07681*.

Pablo Villalobos and 1 others. 2022. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*.

Cédric Villani. 2009. *Optimal transport: old and new*, volume 338. Springer.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023a. Data selection for language models via importance resampling. *arXiv preprint arXiv:2302.03169*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023b. Doremi: Optimizing data mixtures speeds up language model pretraining. In *NeurIPS*.

Zichun Yu, Spandan Das, and Chenyan Xiong. 2024. Mates: Model-aware data selection for efficient pretraining with data influence models. *Advances in Neural Information Processing Systems*, 37:108735–108759.

Paul L Zador. 1982. Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Transactions on Information Theory*, 28(2):139–149.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Qiu Jiantao, Lei Cao, Ju Fan, Ye Yuan, Guoren Wang, and Conghui He. 2025a. Harnessing diversity for important data selection in pretraining large language models. In *The Thirteenth International Conference on Learning Representations*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Xinlin Zhuang, Jiahui Peng, Ren Ma, Yinfan Wang, Tianyi Bai, Xingjian Wei, Qiu Jiantao, Chi Zhang, Ying Qian, and Conghui He. 2025. Meta-rater: A multi-dimensional data selection method for pre-training language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10856–10896.

A Unsupervised Hyperparameter Derivation

In Eq. (1), the scoring function relies on a weight vector $\mathbf{w}$ to balance diverse geometric features. Instead of relying on heuristic grid search, we derive $\mathbf{w}$ intrinsically from the data geometry. This process is grounded in the statistical stability analysis and spectral consensus visualized in Figure 7 and Figure 6.

1. Feature Stabilization (Log-Normal Priors). Let $\mathcal{F} = \{z_{\text{coh}}, z_{\text{len}}, z_{\text{ent}}, z_{\text{size}}\}$ be the set of raw feature vectors. As visualized in **Figure 7**, raw extensive metrics (Length, Size) exhibit heavy-tailed instabilities. To mitigate this, we first apply log transform to stabilize their magnitudes. We define the **stabilized feature matrix** $\mathbf{Z}^\dagger$ as follows:

$$\begin{aligned} z^\dagger_{\text{coh}} &= z_{\text{coh}} \\ z^\dagger_{\text{ent}} &= z_{\text{ent}} \\ z^\dagger_{\text{len}} &= \log(z_{\text{len}}) \\ z^\dagger_{\text{size}} &= \log(z_{\text{size}}) \end{aligned} \tag{12}$$

Note that at this stage, $z^\dagger$ represents the raw physical properties (e.g., larger $z^\dagger_{\text{len}}$ still means longer sequence).

2. Standardization and Main Text Notation. We then apply Z-score standardization to the entire matrix $\mathbf{Z}^\dagger$ to ensure all dimensions share a unified scale (zero mean, unit variance). This yields the normalized metrics $\tilde{\mathbf{z}}$ utilized in the main text (Eq. 1):

$$\tilde{z}_{k,f} = \text{Z-Score}(z^\dagger_{k,f}) \tag{13}$$

Thus, $\tilde{z}$ preserves the original polarity of the features. This is why Eq. 1 explicitly subtracts the penalty terms (Length, Entropy, Size) to convert them into a quality score.

3. Spectral Weight Derivation. To derive the consensus weights $\mathbf{w}$, we construct a temporary **Aligned Matrix** $\mathbf{X}_{align}$ where all features are oriented towards "quality" (flipping the signs of terms):

$$\mathbf{X}_{align} = [\tilde{z}_{\text{coh}}, -\tilde{z}_{\text{ent}}, -\tilde{z}_{\text{len}}, -\tilde{z}_{\text{size}}] \tag{14}$$

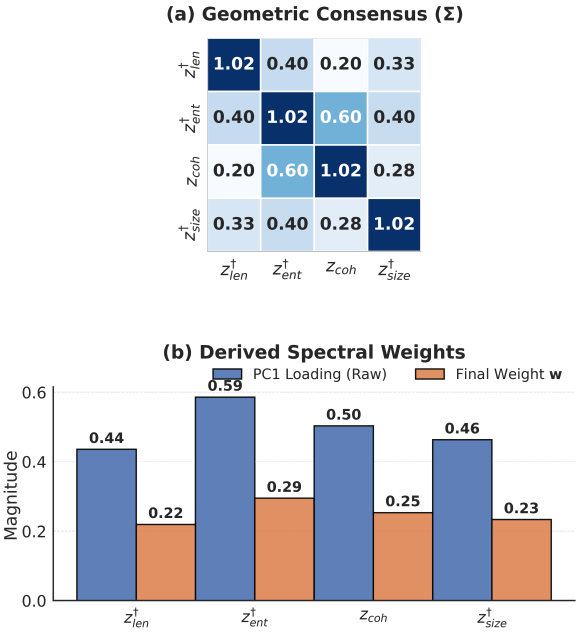

Figure 6: **Unsupervised Spectral Weight Derivation.** **(a)** The geometric consensus matrix $\mathbf{\Sigma}$, computed on the **polarity-aligned metrics** (where penalties are inverted), reveals strong positive correlations (e.g., 0.60 between aligned entropy and cohesion). This suggests that these proxies share a common latent direction. **(b)** Instead of heuristic tuning, we derive the balancing coefficients $\mathbf{w}$ directly from the first principal component (PC1). The spectral analysis intrinsically yields data-driven weights in which **Entropy** and **Cohesion** receive the largest coefficients.

We then compute the covariance matrix $\mathbf{\Sigma} = \frac{1}{K-1}\mathbf{X}^\top_{align}\mathbf{X}_{align}$. As shown in **Figure 6(a)**, this alignment reveals strong positive correlations. The first principal eigenvector $\mathbf{v}_1$ of $\mathbf{\Sigma}$ captures the direction of **Maximum Consensus**. The final weights are derived by L_1-normalizing this vector: $\mathbf{w} = \mathbf{v}_1 / \|\mathbf{v}_1\|_1$.

B Intrinsic Resolution Selection Details

B.1 Metric Definition: Rank Stability

To robustly quantify the topological stability $J_{stab}(K)$ (rank stability across resolutions), we utilize Kendall's Rank Correlation Coefficient (τ). This metric evaluates whether the relative quality ranking of clusters remains consistent after projecting to a finer resolution.

Given the two scoring vectors $\mathbf{s}^{(K)}$ and $\hat{\mathbf{s}}^{(K)}$ (from Eq. 3), we examine all possible pairs of clusters (i, j) where $1 \le i < j \le K$. A pair is classified based on the consistency of their relative

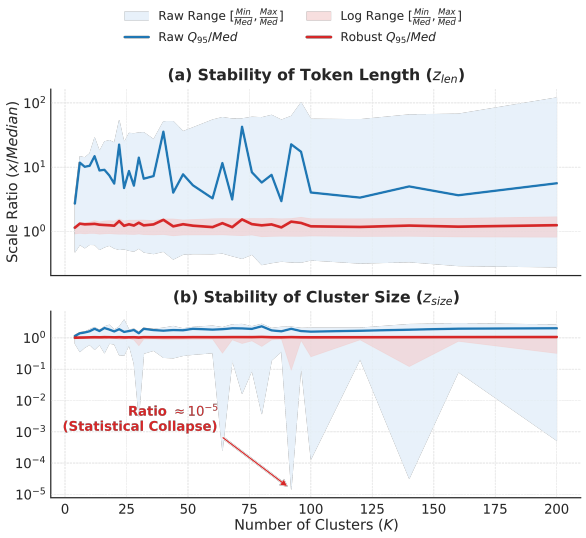

Figure 7: **Statistical Stability and the Necessity of Log Transform.** We analyze the stability of feature statistics across varying cluster resolutions ($K \in [5, 200]$). **(a)** For token length, the raw scale (blue) exhibits high-frequency oscillations, while our **log-stabilized estimator** ($z^{\dagger}_{len}$, red) remains smooth. **(b)** For cluster size, the raw statistics suffer from extreme **statistical collapse** (highlighted by the arrow, where the **minimum-to-median ratio** drops to $\sim 10^{-5}$), which would cause numerical instability in standard Z-score calculations.

ordering:

$$\text{Concordant} : (s_i - s_j)(\hat{s}_i - \hat{s}_j) > 0 \tag{15}$$

$$\text{Discordant} : (s_i - s_j)(\hat{s}_i - \hat{s}_j) < 0 \tag{16}$$

Let N_{conc} and N_{disc} denote the total counts of such pairs:

$$N_{\text{conc}} = \sum_{i<j} \mathbb{I}(\text{Concordant})$$

$$N_{\text{disc}} = \sum_{i<j} \mathbb{I}(\text{Discordant})$$

where $\mathbb{I}(\cdot)$ is the indicator function. The final stability metric is the normalized difference between these counts:

$$J_{stab}(K) = \frac{N_{conc} - N_{disc}}{K(K-1)/2} = \frac{2(N_{conc} - N_{disc})}{K(K-1)} \tag{17}$$

The denominator $K(K-1)/2$ represents the total number of unique pairs. Thus, $J_{stab} \approx 1$ indicates that the geometric structure is perfectly preserved across resolutions.

B.2 Implementation Optimizations

To further ensure stability against sample size variations, we implement the following engineering optimizations:

1. Multi-Scale Hop Averaging. Instead of relying solely on the immediate neighbor ($K \to K+1$), which may be noisy, we compute stability across multiple strides $\Delta K \in \{2, 4, 6\}$. The final stability is a weighted average:

$$J_{final}(K) = \frac{\sum_{h \in \Delta K} \gamma_h \cdot J_{stab}(K \to K+h)}{\sum \gamma_h} \tag{18}$$

where γ_d are decay weights (e.g., $[0.5, 0.3, 0.2]$) that prioritize local consistency.

2. Small-Sample Fisher Shrinkage. When K is small, rank-correlation estimates can have high variance and appear overly optimistic. We therefore apply an **atanh-based shrinkage heuristic** to damp inflated stability scores in low-K regimes. We treat $J_{stab} \in (-1, 1)$ as a bounded rank-stability score and map it with $z = \operatorname{arctanh}(J_{stab})$ for shrinkage; this is an engineering correction rather than a statistical guarantee.

$$z_{shrunk} = z \cdot \tanh\left(\lambda_{shrink} \cdot \sqrt{N_{valid} - 3}\right), \tag{19}$$

$$r_{\text{shrunk}} = \tanh(z_{\text{shrunk}}). \tag{20}$$

where N_{valid} denotes the effective number of clusters participating in the rank comparison and λ_{shrink} is a regularization parameter.

C Theoretical Proofs

In this section, we provide the detailed derivation for Theorem 1. We model the data selection process as an optimal quantization problem on a Riemannian manifold, drawing connections to recent theoretical advances in data pruning (Sorscher et al., 2022). All bounds below hold for an arbitrary number of clusters K; in UniGeM we instantiate $K = K^*$, where K^* is selected by maximizing the stability objective $J_{stab}(K)$ (Algorithm 1).

C.1 Proof of Error Decomposition

Let $Q : \mathcal{M} \to \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ be the quantization operator mapping any point x to its nearest global centroid $\mathbf{c}_{k(x)}$. Let C_k denote the Voronoi cell induced by $\mathbf{c}_k$, and define $\alpha_k \triangleq \mu(C_k)$.

Starting from the definition of the squared Wasserstein-2 distance (Peyré et al., 2019),

$$\mathcal{E}(S) = W_2^2(\mu, \hat{\mu}_S)$$

$$= \inf_{\gamma \in \Pi(\mu, \hat{\mu}_S)} \int_{\mathcal{M} \times \mathcal{M}} \|x - y\|^2 \, d\gamma(x, y) \tag{21}$$

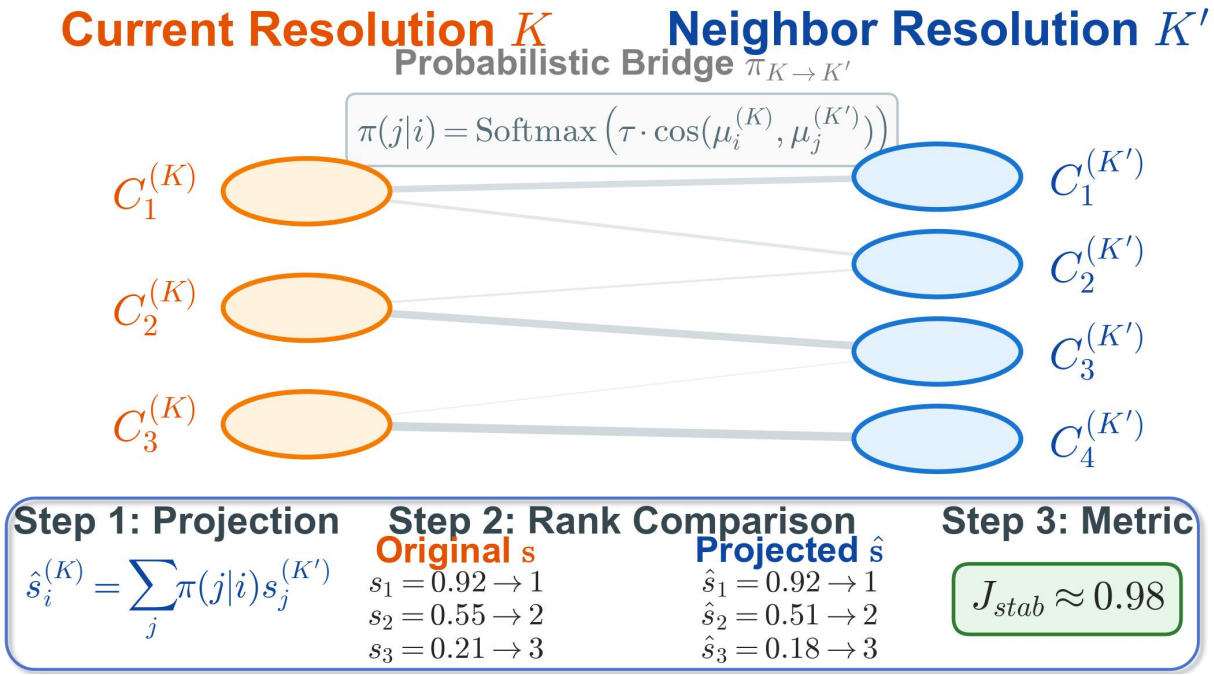

Figure 8: **Schematic of Intrinsic Resolution Selection.** The workflow proceeds in three stages corresponding to the mathematical derivation: (1) **Projection:** A probabilistic bridge π is constructed via centroid similarity (Eq. 2) to back-project scores from neighbor resolution K', yielding the reconstructed profile $\hat{s}^{(K)}$ (Eq. 3); (2) **Rank Comparison:** The relative ordering of the original geometric scores $\mathbf{s}^{(K)}$ is compared against the reconstructed proxies $\hat{\mathbf{s}}^{(K)}$; (3) **Metric Computation:** The final stability score J_{stab} is derived using Kendall's τ rank correlation (Eq. 4). Numerical values (e.g., 0.92) are schematic examples for illustration.

we upper bound $\mathcal{E}(S)$ by constructing an explicit two-stage transport plan through a centroid-supported intermediate measure. Define $\mu_K \triangleq \sum_{k=1}^{K} \alpha_k \, \delta_{\mathbf{c}_k}$. By the triangle inequality of W_2,

$$W_2(\mu, \hat{\mu}_S) \le W_2(\mu, \mu_K) + W_2(\mu_K, \hat{\mu}_S), \quad (22)$$

and using $(a+b)^2 \le 2a^2 + 2b^2$ yields the constant-factor bound

$$\mathcal{E}(S) \le 2W_2^2(\mu, \mu_K) + 2W_2^2(\mu_K, \hat{\mu}_S). \quad (23)$$

Stage-I (Global distortion). Couple each $x \sim \mu$ with its centroid $Q(x)$ to obtain

$$W_2^2(\mu, \mu_K) \le \sum_{k=1}^{K} \int_{C_k} \|x - \mathbf{c}_k\|^2 \, d\mu(x) \triangleq \mathcal{T}_1. \quad (24)$$

Stage-II (Within-cluster residual). For analysis, we view the selected-set empirical measure as a cluster-wise mixture

$$\hat{\mu}_S \triangleq \sum_{k=1}^{K} \alpha_k \, \hat{\mu}_{S_k}, \quad (25)$$

where $\hat{\mu}_{S_k}$ is the empirical measure supported on S_k. Let $S_k \triangleq S \cap C_k$ be the selected subset inside cluster C_k. The second term $W_2^2(\mu_K, \hat{\mu}_S)$ measures how well the selected points within each C_k represent the local mass anchored at $\mathbf{c}_k$. This induces a within-cluster residual term that we summarize by

$$W_2^2(\mu_K, \hat{\mu}_S) \le \sum_{k=1}^{K} \alpha_k \, \mathbb{V}(S_k) \triangleq \mathcal{T}_2. \quad (26)$$

where we define the within-cluster *residual energy* (a second moment w.r.t. the centroid) as

$$\mathbb{V}(S_k) \triangleq \mathbb{E}_{x \sim \hat{\mu}_{S_k}} \|x - \mathbf{c}_k\|^2. \quad (27)$$

Combining the two parts with Eq. (23) yields a constructive decomposition of $\mathcal{E}(S)$ into a global quantization term $\mathcal{T}_1$ and a local within-cluster term $\mathcal{T}_2$, up to constant factors commonly used in quantization-style analyses (Gray and Neuhoff, 2002).

Remark (High-dimensional intuition). In high-dimensional embeddings ($d \gg 1$), cross-terms between centroid error and within-cluster residual are often empirically small due to concentration effects (Ledoux, 2001), which motivates the near-additive behavior observed in practice; however, our bound above does not rely on this approximation.

C.2 Bound Derivation for Stage-I ($\mathcal{T}_1$)

The first term $\mathcal{T}_1$ corresponds to the classical high-resolution quantization error. According to **Zador's Theorem** (Zador, 1982), for a quantizer with K codepoints on a d-dimensional manifold with probability density function $p(x)$, the asymptotic distortion satisfies:

$$\lim_{K\to\infty} K^{2/d} \cdot \mathcal{T}_1 = J_d \|p\|_{d/(d+2)}, \tag{28}$$

where J_d is the coefficient of the optimal lattice quantizer in $\mathbb{R}^d$. This yields the Stage-I term in our bound:

$$\mathcal{T}_1 \le C_d \cdot K^{-2/d}. \tag{29}$$

Implication: This suggests that Stage-I controls global covering distortion through the choice of K.

C.3 Bound Derivation for Stage-II ($\mathcal{T}_2$)

The second term $\mathcal{T}_2$ represents the intra-cluster residual variance. For a standard random sampler, this corresponds to the raw cluster variance, which we denote by σ_k^2 for cluster C_k. Let $p_k(x) \triangleq p(x)/\alpha_k$ for $x \in C_k$ denote the conditional density within cluster C_k, where $\alpha_k = \mu(C_k)$.

Assumption (dominant structural filtering). Within each cluster C_k, we assume the auxiliary reweighting terms used in practice (e.g., probe score P_{S_j} and cohesion gate β_{S_j}) are either (i) approximately independent of the radial deviation $\|x - \mathbf{c}_k\|$ or (ii) bounded and do not systematically favor higher-deviation points. Under this assumption, the dominant geometric effect of Stage-II is governed by $\exp(-\lambda \mathcal{L}_{struct}(x))$.

UniGeM modulates the sampling probability via $P(x) \propto \exp(-\lambda \mathcal{L}_{struct}(x))$. For theoretical analysis, we approximate this soft exponential decay as a truncation mechanism on an effective acceptance region

$$\Omega_{UniGeM} = \{x \in C_k \mid \mathcal{L}_{struct}(x) < \tau\}, \tag{30}$$

where τ is a confidence threshold implicitly controlled by λ.

Define the random-baseline within-cluster second moment as

$$\sigma_k^2 \triangleq \int_{C_k} \|x - \mathbf{c}_k\|^2 p_k(x)\, dx. \tag{31}$$

Under the truncation approximation, UniGeM induces the conditional density $q_k(x) \triangleq \frac{p_k(x)\mathbf{1}[x\in\Omega_{UniGeM}]}{Z_k}$ with $Z_k \triangleq \int_{\Omega_{UniGeM}} p_k(x)\, dx$, and the resulting second moment is

$$\mathbb{V}(S^{(k)}_{UniGeM}) \approx \int_{\Omega_{UniGeM}} \|x - \mathbf{c}_k\|^2 q_k(x)\, dx. \tag{32}$$

We define the pruning gain as

$$\Delta^{(k)}_{gain} \triangleq \sigma_k^2 - \int_{\Omega_{UniGeM}} \|x-\mathbf{c}_k\|^2 q_k(x)\, dx \quad \ge 0, \tag{33}$$

where the non-negativity holds when the acceptance region preferentially keeps lower-deviation points.

C.4 Final Theorem Assembly and Remark

Substituting the bounds for $\mathcal{T}_1$ and $\mathcal{T}_2$ back into the decomposition yields:

$$\mathcal{E}(S_{UniGeM}) \le 2C_d K^{-2/d} + 2\sum_{k=1}^{K} \alpha_k \big(\sigma_k^2 - \Delta^{(k)}_{gain}\big). \tag{34}$$

Practical proxy. In Eq. (11), the gain $\Delta^{(k)}_{gain}$ captures the variance mass removed by pruning within C_k. In practice, high-deviation samples correspond to large standardized feature deviations; thus $\mathcal{L}_{struct}(x)$ (a rectified squared Mahalanobis distance) directly serves as a proxy for identifying the rejected region Ω^c and monitoring pruning strength.

D Annotation Model.

We leverage the Qwen3-235B model as a **Knowledge Probe** to inspect the semantic attributes of these samples. Adopting a **Model-Based Annotation** strategy (Seed et al., 2025), we design a structured system prompt to decompose the analysis into four complementary dimensions: *Code Quality*, *Engineering Design*, *Training Suitability*, and *Knowledge Density*. This multi-dimensional rubric aims to capture both syntactic correctness and training-relevant content.

The specific system prompt used for this annotation is provided below.

System Prompt: Code Data Evaluation Strategy

You are an expert code evaluator. Your task is to assess the provided code snippet based on four distinct dimensions. For each dimension, assign a score on a scale of 1 to 5 according to the criteria below.
1. Code Quality & Compliance (Focus: Syntax, nam-

ing, and readability)
- **1-2 (Low):** Severe syntax errors or logical flaws; chaotic naming conventions; inconsistent indentation; contains dead code or empty functions; non-executable.
- **3-4 (Mid):** Syntactically correct and executable; reasonable naming and formatting; minor redundancies or loose structure but generally readable.
- **5 (High):** Error-free; strict adherence to standards (e.g., PEP8); precise naming; concise logic with no redundancy; highly readable and linear structure.

2. Algorithmic & Engineering Design (Focus: Modularity, robustness, and system thinking)
- **1-2 (Low):** Monolithic structure (global scope); lack of modularity; absence of input validation or exception handling; hardcoded constants; fails on edge cases.
- **3-4 (Mid):** Basic modularity (function splitting); separation of concerns; foundational error checking; clear structure but lacks advanced abstraction or extensibility.
- **5 (High):** High-level abstraction (classes/patterns); robust engineering (exception safety, resource management); extensible, testable, and demonstrates system-level thinking.

3. Training Suitability (Focus: Educational value and style consistency for LLMs)
- **1-2 (Low):** Arbitrary naming; missing or misleading comments; mixed styles; hallucinated or fragmentary code; lacks context.
- **3-4 (Mid):** Normative naming; comments cover key steps; consistent style; readable logic suitable for beginners or general training.
- **5 (High):** Strict adherence to community best practices; insightful comments explaining design intent; exemplary structure suitable for teaching or high-quality fine-tuning.

4. Knowledge Density (Focus: Technical insights and domain expertise)
- **1-2 (Low):** Trivial operations (e.g., simple I/O, loops); lacks domain knowledge; low information density equivalent to introductory tutorials.
- **3-4 (Mid):** Standard algorithms or patterns (e.g., DFS, HashMaps); implements common best practices; explicit but generic knowledge.
- **5 (High):** Non-trivial insights or cross-domain knowledge (e.g., memory alignment, bitwise optimization, lock-free structures); reveals underlying principles or deep optimizations.

Output Format: Output strictly in valid JSON format: `{"code_quality": int, "algorithm_and_engineering": int, "training_suitability": int, "knowledge_score": int}`
Data Input: $content

E Experiment Details

In this appendix, we provide the condensed technical specifications for our experiments, including the model architecture, training recipe, and adaptation protocols for SOTA baselines.

E.1 Model Architectures

We employ a fine-grained sparse MoE architecture with identical expert parameterization across models. We scale total capacity by increasing the number of experts, while keeping the routing strategy fixed (Top-2) so that the per-token activated parameter budget remains constant, preserving inference throughput.

Configuration	UniGeM-8B	UniGeM-16B
Total / Active Params	8.0B / 1.4B	16.8B / 1.4B
Total Experts (N)	32	64
Routing Strategy	Top-2	Top-2
Hidden / Layers	2048 / 24	2048 / 24

E.2 Pre-training Configuration

The models are trained from scratch on a 100B-token mixture, consisting of 70B code tokens and 30B code-related text tokens. We utilize **Qwen3-235B** to retrieve the text component from Common Crawl to ensure semantic relevance. The training employs a Warmup-Stable-Decay (WSD) schedule.

Hyperparameter Selection Logic. The geometric hyperparameters $\{\lambda, T_{scale}, \epsilon\}$ are calibrated based on the statistical moments of the 20% probe manifold. We set the **structural** $\lambda = 0.5$ to ensure that samples deviating beyond 2σ from the geometric consensus incur a significant weight reduction (e.g. reduced to $\approx 13.5\%$ of the original weight), effectively pruning logical outliers. The **scale factor** $T_{scale} = 20$ is employed to amplify the density contrast during cross-resolution alignment, ensuring the stability-driven clustering captures sharp manifold boundaries. Finally, an **exploration floor** $\epsilon = 0.01$ is maintained to preserve long-tail distributional diversity and mitigate manifold approximation errors.

Hyper-parameter	Value
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$, WD=0.1)
Peak Learning Rate	3.0×10^{-4} (8B) / 2.4×10^{-4} (16B)
Batch Size / Seq Len	2,560 $\rightarrow$ 8,960 / 4,096 Tokens
Stability Mechanisms	NormHead, Stochastic Routing Warmup
UniGeM Geometric Mining Params	
Structural (λ)	0.5 (calibrated to 2σ consensus)
Exploration Floor (ϵ)	0.01 (1% diversity reserve)
Transition Scale (T_{scale})	20 (Eq. 2)
Clustering & Embedding	
Embedding Model	*Qwen3-embedding*
Global Resolution (K^*)	72 (via stability-driven selection)
K-means Iterations	10 (Stage-I) / 5 (Stage-II)

E.3 SOTA Baseline Adaptation

Standard implementations of CLIMB and Meta-rater are designed for general text and often rely on proxy tasks (e.g., MMLU) or generic encoders (e.g., BERT), which can introduce domain mismatch for code. Using these methods strictly off-the-shelf would make the comparison less informative. We therefore apply a **Code Adaptation** protocol to improve their alignment with code and obtain strong code-aware baselines.

- **Nemotron-CLIMB Adaptation:** We replace general BERT embeddings with **Qwen3-embedding** to perform code-aware semantic clustering ($K = 100$). The optimization target is shifted from MMLU to a **Code Oracle (V_{code})**, which computes weighted validation loss on **MBPP-Sanitized**, **HumanEval-Pack**, and logic-dense samples from **DS-1000**.
- **Meta-rater Adaptation:** We re-define the PRRC framework into **Code-PRRC**, focusing on *Professionalism* (complexity), *Readability* (style), *Reasoning* (flow density), and *Cleanliness* (syntax). We score 500k seed samples via **Qwen3-235B** to distill four specialized **ModernBERT-base** scorers capable of handling long-context code quality assessment.
- **Common Proxy Setup:** To search for optimal weights, both methods utilize a **350M Dense Transformer** as a proxy model. These proxies are trained on 2B token slices across multiple iterations (64 trials for CLIMB; 256 for Meta-rater) to fit a LightGBM-based quality-to-loss regressor.

F Distributional Analysis of Features for Stage-I Cluster

In this section, we provide a detailed distributional analysis of the geometric proxies observed in our large-scale pre-training experiment. Specifically, we visualize the feature statistics across the optimal resolution of $K^* = 72$ **global clusters**, identified via the Topological Stability analysis in Section 4.3. These empirical results validate the statistical assumptions underpinning our Geometric Scoring function (Eq. 1).

Transformation of Extensive Properties. As illustrated in the top row of Figure 9, the raw distributions of **Cluster Size** (z_{size}) and **Sequence Length** (z_{len}) exhibit significant right-skewness across the 72 latent domains. This confirms our hypothesis in Section 2.2 that the raw code corpus is highly heterogeneous, spanning multiple orders of magnitude. Direct usage of these raw metrics would result in *variance dominance*, where spectral analysis is biased by magnitude outliers rather than structural quality. However, as shown in the bottom row, applying the logarithmic transformation effectively projects these features into log-space, where the transformed values are closer to Gaussian. This supports the normality assumption behind Z-score standardization and the subsequent spectral consensus step.

Stability of Intensive Properties. Conversely, the intensive properties—**Cohesion** (z_{coh}) and **Entropy** (z_{ent})—naturally display bounded, unimodal distributions across the 72 clusters without transformation. This implies that the experimentally identified clusters are statistically well-formed, maintaining consistent internal densities and semantic purities. Consequently, we validate our hybrid processing strategy: while extensive features require logarithmic dampening to mitigate scale disparities, intensive features can be directly utilized as linear geometric priors to preserve their original sensitivity.

G Distributional Analysis of Semantic Scoring

In this section, we report the statistical characteristics of the semantic scores obtained during the Stage-II exploration. We visualize both the raw discrete ratings from the Annotation Model and the aggregated continuous scores for sub-clusters to provide a comprehensive view of the data quality distribution.

G.1 Probe and Sub-Cluster Score Distributions

Figure 10 presents the distribution of discrete quality ratings (scale 1–5) assigned by the Annotation Model to individual probe samples. The results indicate a pronounced left-skewed distribution across all four evaluation dimensions. Specifically, the *Code Quality* and *Training Suitability* metrics show a high concentration of samples receiving perfect or near-perfect scores ($> 60\%$ rated as 5).

Figure 11 further illustrates the probability density of the aggregated Semantic Score (P_{S_j}) for sub-clusters. Consistent with the probe-level observations, the cluster-level scores are heavily concentrated in the high-value interval $[4.0, 5.0]$. This

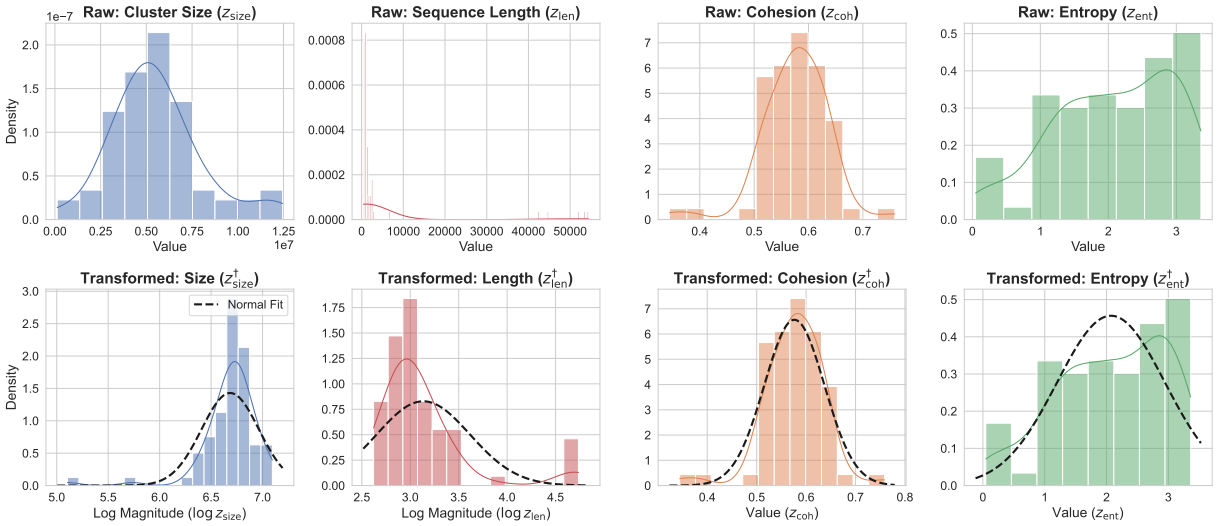

Figure 9: **Distributional Transformation across the $K^* = 72$ Global Clusters. (Top Row)** The raw distributions of extensive properties—*Cluster Size* (z_{size}) and *Sequence Length* (z_{len})—exhibit extreme heavy-tailed skewness across the 72 experimental clusters. **(Bottom Row)** Applying the logarithmic transformation stabilizes these features; their log-values are closer to a Gaussian shape (as illustrated by the overlaid normal fits). In contrast, the intensive properties *Cohesion* and *Entropy* naturally follow a stable unimodal distribution, confirming the robustness of the extracted latent manifolds.

suggests that the majority of the code corpus, after initial filtering, is perceived as syntactically valid and high-quality by the LLM judge.

G.2 Implications for Selection Strategy

The observed "ceiling effect" in the score distributions highlights a potential limitation in relying solely on semantic scoring: the lack of discrimination in the high-score regime. Since a significant portion of sub-clusters achieves a saturated score ($P_{S_j} \approx 5.0$), semantic metrics alone may struggle to differentiate between intrinsic high-value domains and superficially correct samples. This empirical observation motivates the design of the **Structural** ($\mathcal{L}_{struct}$) within the UniGeM framework. By introducing geometric constraints as an orthogonal selection criterion, UniGeM effectively handles these saturated distributions, filtering out sub-clusters that appear high-quality but are topologically inconsistent with the domain manifold.

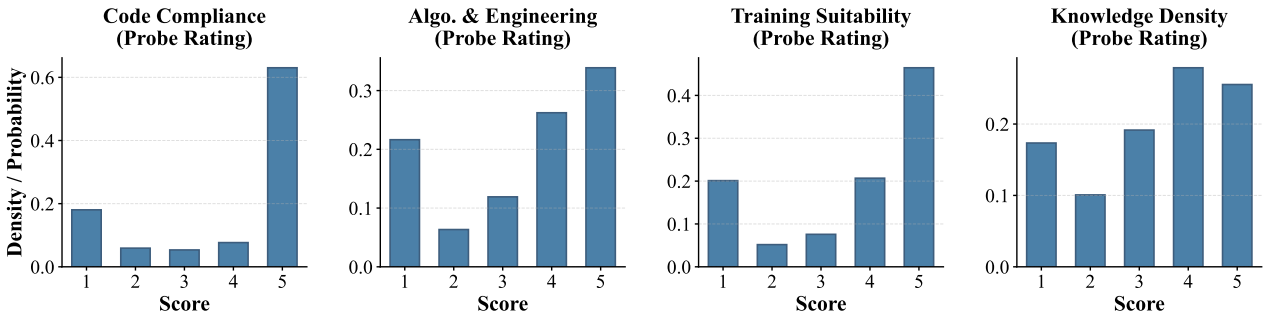

Figure 10: **Discrete Probe Quality Distribution (Raw Prompt Outputs).** The probability mass distribution of integer scores (1–5) assigned by the **Annotation Model** to probe samples. The data exhibits a strong skew towards the upper bound (scores 4 and 5) across all dimensions, reflecting the general high acceptance rate of the judge model for code syntax.

Figure 11: **Sub-cluster Semantic Score Distribution (P_{S_j}).** Kernel density estimation of the aggregated semantic scores for sub-clusters S_j. The distribution shows a saturation effect in the $4.0 - 5.0$ range. This lack of variance in the top quantile underscores the necessity of incorporating geometric penalties ($\mathcal{L}_{struct}$) to introduce discriminative gradients among high-scoring clusters.

H Reproducibility Statement

We provide sufficient details for an independent reimplementation of UniGeM and for reproducing our experimental results.

Method / Algorithm. The full method specification (problem setup, Stage-I Macro-Exploration, Stage-II Micro-Mining, and all scoring terms) is described in Section 2.

Experimental Protocol. The evaluation protocol (corpus construction, sampling ratio, model settings, and compared baselines) is summarized in Section 3. Training and evaluation hyperparameters, together with the calibration logic for the geometric parameters, are provided in Appendix E.

Baselines and Prompts. Details of the code adaptation for CLIMB and Meta-rater are documented in Appendix E. The probe/annotation rubric and the exact system prompt used for semantic scoring are provided in Appendix D.

Code Release. We will release the complete codebase (data processing, clustering/selection, training scripts, and evaluation) in a public repository upon completion of an internal review.

Compute and infrastructure. All pre-training runs were executed on 64 NVIDIA H800 GPUs. Each 100B-token pre-training run took approximately 15 hours wall-clock time. Automated benchmark evaluation was executed on NVIDIA H20 GPUs.

Use of AI assistants. AI assistants were used only for minor language polishing and clarity improvements during writing. All scientific content was developed and verified by the authors.